

Towards Intelligent Control via Genetic Programming

Francesco Marchetti, Edmondo Minisci, Annalisa Riccardi
Intelligent Computational Engineering Laboratory, University of Strathclyde
Glasgow, United Kingdom
{francesco.marchetti,edmondo.minisci,annalisa.riccardi}@strath.ac.uk

Abstract—In this paper an initial approach to Intelligent Control (IC) using Genetic Programming (GP) for access to space applications is presented. GP can be employed successfully to design a controller even for complex systems, where classical controllers fail because of the high nonlinearity of the systems. The main property of GP, that is its ability to autonomously create explicit mathematical equations starting from a very poor knowledge of the considered plant, or just data, can be exploited for a vast range of applications. Here, GP has been used to design the control law in an Intelligent Control framework for a modified version of the Goddard Rocket problem in 3 different failure scenarios, where the approach to IC consists in an online re-evaluation of the control law using GP when a considerably big change in the environment or in the plant happens. The presented results are then used to highlight the potential benefits of the method, as well as aspects that will need further developments.

Index Terms—Genetic Programming, Intelligent Control, Machine Learning, Evolutionary Algorithms, Access to Space.

I. INTRODUCTION

In one of his work [1], Koza, considered the father of GP, pointed out the need and the possibilities related to an automatic methodology to design a controller optimizing both its topology and parameters, so to fully exploit the increasing computational power available to design a controller with better performances and more reliable than classical controllers. This is especially true considering systems like Space Access Vehicles (SAVs), hypersonic vehicles and spacecrafts for space exploration, which are defined by very complex physical models with a high degree of non linearity, and potentially very big uncertainties on operational conditions and models. Also other researchers addressed these problems, for example, as described in [2], traditional control design methodologies are not always sufficient to completely define a complex system such an hypersonic vehicle, hence the need of a new design method incorporated into an IC framework. The need of IC is also pointed out in [3], where traditional control methodologies used in the aerospace industry are analyzed and it is concluded that the controller design phase is deeply affected by the lack of accurate mathematical models and the high-order and nonlinear character of these models, hence the need of a new kind of controller capable of taking into account these issues. To tackle the problem described above, namely non linearities in the models and considerably big uncertainties in the environment or in the control system, GP has been chosen to produce the control law of the considered system within

an IC framework. For these kind of applications GP could be a powerful tool since it allows the automatic creation of a control law, with minimal human intervention. As mentioned above, the application of GP for control system generation is not new in the literature, but it is still not very common, especially for aerospace applications. Some works regard the design of a PID-based controller using GP, as in [4] and [5]. While others focus more on the particular GP algorithm used for the controller design task, rather than on its application to a realistic test case as in [6]. Moreover, up to now, none of the already available works has aimed at designing a controller for aerospace applications only using GP in an IC framework.

Such IC framework is particularly important and necessary if the considered system has to autonomously and successfully cope with uncertainties due to a lack of knowledge about the environment and/or the control system at design time.

The paper is organized as follows: in Section II the theory and some of the most important aspects of GP are presented; in Section III IC is defined and the considered approach to it is described. Section IV describes the selected test case, while in Section V some results obtained over 3 selected failure scenarios are presented. Finally in Section VI the conclusions and some suggestions for future works are given.

II. GENETIC PROGRAMMING

Genetic Programming is a Machine Learning (ML) technique originally introduced by Koza [7] in 1994. It pertains to the class of evolutionary algorithms and it is capable of creating a computer program by following the laws of biological evolution. In the GP, an individual is composed by a tree as depicted in Fig. 1, where each tree represents a mathematical equation itself. During the evolutionary process the nodes of the tree, called genes, are recombined using evolutionary operators such as crossover and mutation, to produce an offspring. The individuals in the population will then be selected in order to maximize or minimize a certain fitness function.

Given these characteristics, GP is well suited for symbolic regression applications such as the design of a controller optimizing both its topology and its parameters, as shown in [1]. In fact, a controller, from the mathematical point of view, is nothing more than a mathematical model that, given certain inputs, produces a desired output. According to this, the extreme flexibility of GP can lead to useful results in control design applications, especially for those problems where high

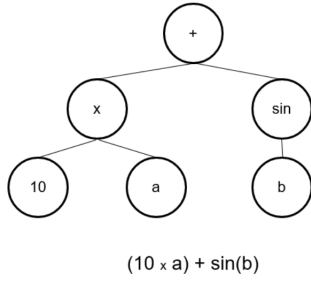


Fig. 1. Individual representation in Genetic Programming

non linearities are present, thus impairing or seriously limiting the use of classical controllers.

Moreover, the ability to produce a mathematical equation is especially important in comparison to other ML techniques, since such equation could be analysed by the user with well known techniques instead of having only a black box model. Unfortunately, GP's high flexibility comes with a great computational cost compared to other ML techniques, such as Artificial Neural Networks (ANNs), and there are still some open issues which impair the creation of a generalized methodology for different GP applications.

Among these issues, there are:

- Constraints Handling: how infeasible individuals are recognized and treated.
- Exploration vs. Exploitation: relation between crossover and mutation rates, and other techniques to avoid local minima.
- Diversity: how to improve and maintain population diversity to avoid local minima and increase the chance of exploring all the search space.

The constraints handling problem is thoroughly analysed in [8], where many different constraints handling techniques are presented, but the author also says how difficult if not impossible is to create a comprehensive technique which can be applied with minor modifications to all kinds of problems. In the test case considered for this work, just few constraints are imposed and yet the GP is not always able to find a feasible solution and must be guided to feasible regions of the search space. In addition, it can also happen that these feasible regions do not contain the global minima, thus requiring a more complex approach to solve this problem.

The exploration vs. exploitation dualism is well described in [9] and, as for the constraint handling problem, there is no unique solution to this issue since there can be many different approaches according to the particular problem considered. About diversity handling, more can be found in [10].

III. INTELLIGENT CONTROL

Intelligent Control has been discussed and studied since many decades, but great confusion lingers around this term. In fact, many works were published in the past decades but after a more in-depth analysis performed for the European Space

Agency (Contract Number: 4000124916/18/NL/CRS/hh), it has been realized that not all of the published works on IC can be actually described as IC according to its standard definition. Such definition comes from both the work of Saridis [11] and Antsaklis [12], and can be summarized as follows: *a controller can be defined intelligent if it can deal autonomously with unforeseen changes in the environment, in the control system or in the goals, by relaying on techniques pertaining to the fields of Artificial Intelligence, Operations Research and Automatic Control.*

According to this definition the authors also proposed a taxonomy [13], and according to such taxonomy the IC approach presented in this paper classifies as E3 C4 G0, which means that the proposed controller can autonomously deal with an *Underlying physics of environment not well defined (E3)*, *No known controller structure (C4)*, and with *Goals entirely pre-determined by designer (G0)* at design time.

A. Proposed approach to IC using GP

For this first step towards Intelligent Control, GP is used to perform the online evaluation of the control law every time a change in the environment or in the considered plant happens. To do so, each of the 2 control parameters, the radial and tangential Thrust, has been divided into two parts as shown in Eq. 1. The first part, $T_{ref}(t)$, is fixed and it is the reference value of the thrust obtained from the optimal trajectory process. This term is useful since it provides the GP some knowledge about the plant and facilitates the evaluation of the second term, which is composed by the control law evaluated with GP every time the tracking error between the states values and their reference values becomes greater than a certain threshold.

$$T(t) = T_{ref}(t) + f_{GP}(e_1(t), \dots, e_n(t)) \quad (1)$$

A main issue arise with this approach, namely the initial conditions for the GP evaluation must be defined *a priori*. That means, that for the considered problem which requires a propagation of a dynamical system, the time used by the GP to perform the evaluation must be defined *a priori* as well. Such time interval must be defined in a conservative manner so to be slightly bigger than the actual time required by the GP, but small enough to allow the controller to effectively control the plant. Due to the stochastic nature of GP, the control law was not always successfully evaluated within the given time interval, so to justify the applicability of the proposed approach to IC some statistics regarding how many times the evaluation was successfully performed within the fixed time interval are presented in Subsec.V-D.

IV. TEST CASE

The chosen test case is a 2-dimensional version of the Goddard Rocket problem [14]. The mathematical model is shown in Eqs. 3, where the thrust components T_r and T_t are the control parameters and the drag components are defined as follows:

$$D_r = \frac{1}{2} \rho v_r \sqrt{v_r^2 + v_t^2} C_d S$$

$$D_t = \frac{1}{2} \rho v_t \sqrt{v_r^2 + v_t^2} C_d S$$

The density model used to obtain the optimal trajectory is:

$$\rho = \rho_0 e^{-\beta r} \quad (2)$$

$$\begin{cases} \dot{r} = v_r \\ \dot{\theta} = \frac{v_t}{r} \\ \dot{v}_r = \frac{T_r}{m} - \frac{D_r}{m} - g + \frac{v_t^2}{r} \\ \dot{v}_t = \frac{T_t}{m} - \frac{D_t}{m} + \frac{v_r v_t}{r} \\ \dot{m} = -\frac{\sqrt{T_r^2 + T_t^2}}{g_0 I_{sp}} \end{cases} \quad (3)$$

The plant constant parameters used for this problem are: C_d = Drag coefficient = 0.6, S = Reference surface = 4.0 m^2 , I_{sp} = Specific impulse = 300 s, m_0 = Total initial mass = 100000 kg, m_p = Propellant fraction of the total mass = 0.99* m_0 , β = Scale factor = 1/8500 m^{-1} , ρ_0 = Air density at sea level = 1.225 $\frac{kg}{m^3}$, g_0 = Gravitational acceleration at sea level = 9.80665 $\frac{m}{s^2}$, R_e = Earth radius = 6371 km.

And the gravitational acceleration is evaluated as follows:

$$g = g_0 \left(\frac{R_e}{r} \right)^2$$

The chosen mission profile, used to obtain the optimal trajectory, consists of reaching an altitude of 400 km while minimizing the propellant consumption. The optimal trajectory, consisting of both the states and controls values, was obtained with a Direct Pseudospectral Collocation method [15]. The control scheme used in non-nominal conditions (Closed Loop) is depicted in Fig. 2. For nominal conditions (Open Loop), only the reference values of the Thrusts were used in order to have an open loop controller. The control laws used are listed in Tab. I. The closed Loop control laws, it was chosen to make the radial (tangential) component of the thrust only dependant on the tracking errors of the radial (tangential) quantities, this to simplify the GP evaluation and reduce the complexity of the control equations.

TABLE I
OPEN LOOP AND CLOSED LOOP CONTROL LAWS

Open Loop	Closed Loop
$T_r(t) = T_{r_{ref}}(t)$	$T_r(t) = T_{r_{ref}}(t) + f_{GP_r}(e_r(t), e_{v_r}(t))$
$T_t(t) = T_{t_{ref}}(t)$	$T_t(t) = T_{t_{ref}}(t) + f_{GP_t}(e_\theta(t), e_{v_t}(t))$

V. RESULTS

The code was developed in Python 3 and it relies on the open source library DEAP [16] for the GP part. All the simulations were run on a Desktop PC with 8GB of RAM and an Intel®Core™ i7-6700 CPU @3.40 GHz x 8 processors and multiprocessing was used. The used GP algorithm has been set as follows:

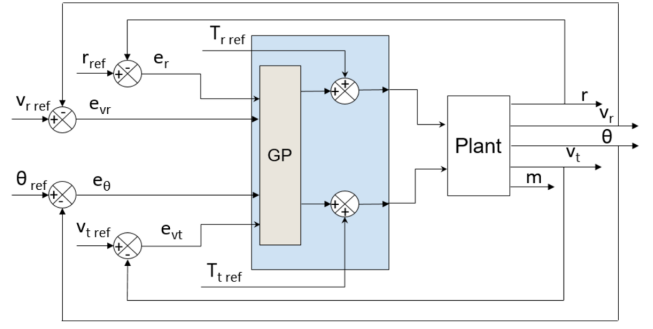


Fig. 2. Simplified Control scheme of modified Goddard Rocket for non-nominal flight conditions

- **Primitives:** +, ++, -, *, tanh, √, log, exp, sin, cos. All the primitives are modified in order to avoid numerical errors.
- **Fitness Function:** two objectives are simultaneously considered
 - 1) $\min fitness_1 = \int |e_r(t)| dt$
 - 2) $\min fitness_2 = ||constraints_violation||_2$
- **Termination Criteria:** the evolution process is stopped if
 - $fitness_1 \leq 0.7$ and $fitness_2 = 0$, or
 - $generation_number = 150$
- **Crossover Rate:** 0.2
- **Mutation Rate:** 0.7
- **Elitism Rate:** 0.1

The mutation rate is set at 0.7 at the beginning of the evolution process in order to explore the search space. Then, when feasible individuals are found ($fitness_2 = 0$), the mutation rate is decreased by 0.01 and the crossover rate is increased by the same quantity at each generation, until the crossover rate reaches the value of 0.65. It is not the aim of this work to focus on the peculiarities of the used GP algorithm, hence a pseudo-code is not provided. The code used in this work can be found at <https://github.com/strath-ace/smart-ml>.

The approach to IC has been tested on 3 different failure scenarios representative of real world situations:

- 1) a sudden failure of a vehicle component, simulated as a variation of one of the plant parameters;
- 2) an unforeseen change of the environmental conditions, through the insertion of a wind gust, which is something that can heavily affect the performances of a SAV;
- 3) poor knowledge of the physical models of the environment at design phase, which is representative, for example, of space exploration applications where many times the environmental models that the spacecraft or the lander will encounter during its mission are not exactly defined, but the designers have to use environmental models obtained through more or less precise observations.

More on these scenarios is explained in the following.

All figures, from Fig. 3 to Fig. 17, show the altitude r and the path angle θ profiles, which are the main quantities of interest.

Moreover, the mission is considered successful if the final values of the altitude and the θ angle are within 1% range from the reference values.

A. 1st Scenario - C_d variation

In this scenario, a sudden variation in one of the system parameters is simulated by varying the drag coefficient C_d from its nominal value 0.6 to a random value at a random time. The C_d random value is chosen in the interval $[0.61, 2]$, while the random change time is chosen in the interval $[20, 250]$ s.

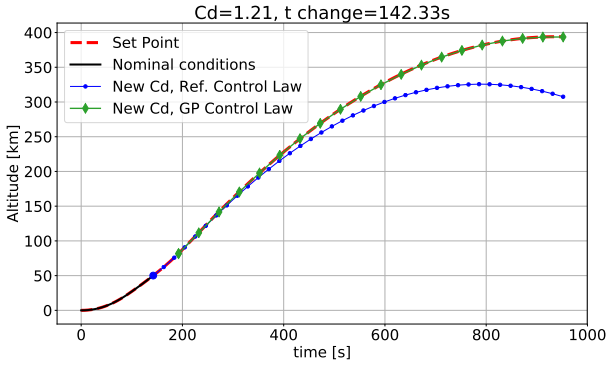


Fig. 3. Altitude profiles for a C_d variation from 0.6 to 1.21 at 142.33 s

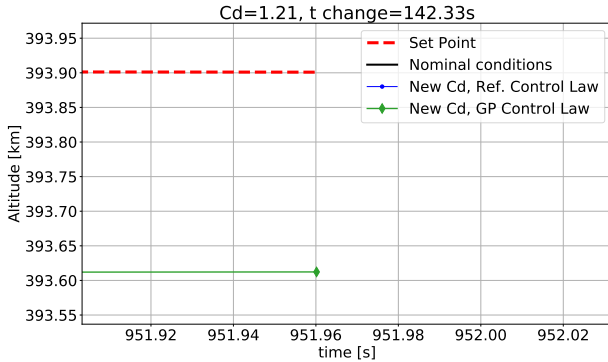


Fig. 4. Detail of the last part of the altitude profile

Figures from 3 to 6 show how a change in C_d from 0.6 to 1.21 at 142.33 s produces a disturbance for the system which is not able to fulfill the mission (dotted line in Fig. 3). While, with a control law evaluated through GP the system can successfully fulfill its mission (diamond marked line), with a difference of about 300 m between the desired final position and the reached final position, as in Fig. 4. Regarding θ , its final value is not deeply affected by the C_d variation, as can be seen from Fig. 5. Nonetheless, Fig. 6 shows that the final θ value using the GP law is closer to the reference than the one reached with the open loop control.

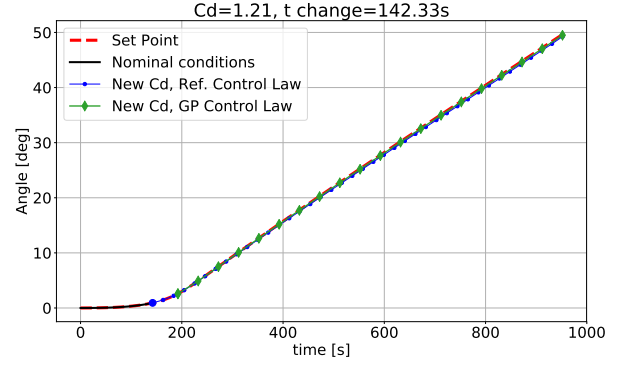


Fig. 5. θ profile for a C_d variation from 0.6 to 1.21 at 142.33 s

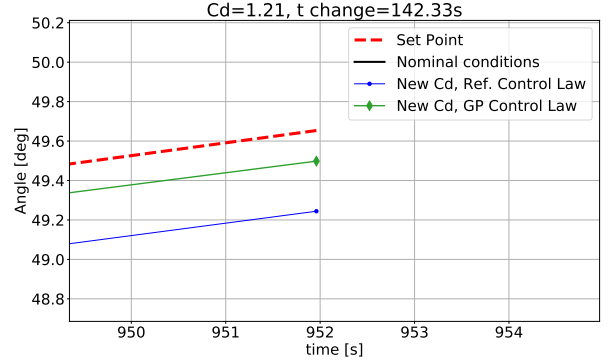


Fig. 6. Detail of the last part of the θ profile

B. 2nd Scenario - Wind Gust

In this scenario, an unpredicted variation of the environmental conditions has been modelled as a wind gust in a random altitude range, with a random constant speed acting in horizontal direction. The beginning of the altitude range for the wind gust was chosen in the interval $[0, 40]$ km and its size was chosen in the interval $[10, 15]$ km. The wind speed was chosen in the interval $[0, 24]$ m/s.

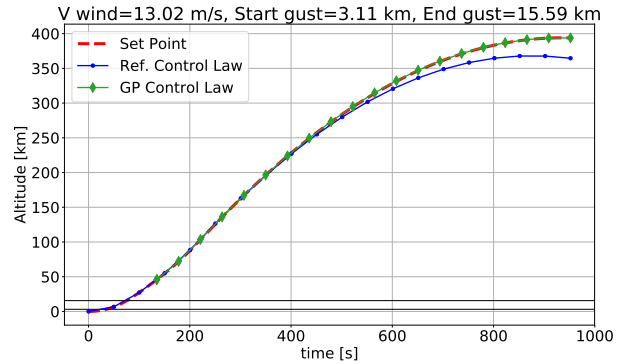


Fig. 7. Altitude profiles for a wind gust of 13.02 m/s applied between 3.11 km and 15.59 km

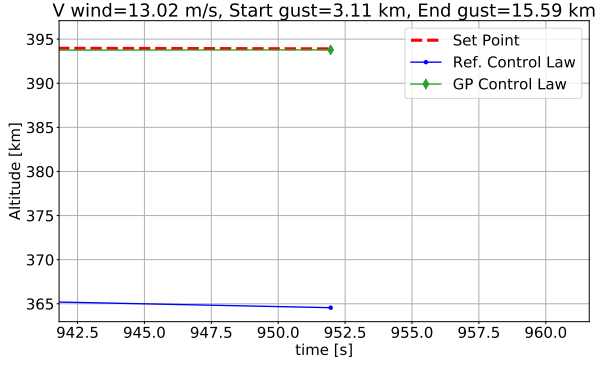


Fig. 8. Detail of the last part of the altitude profile for a wind gust of 13.02 m/s applied between 3.11 km and 15.59 km

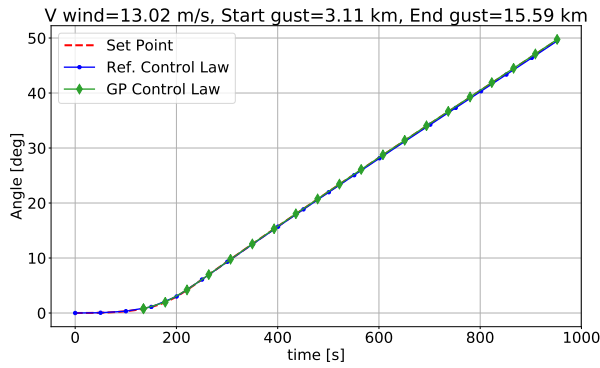


Fig. 9. θ profile for a wind gust of 13.02 m/s applied between 3.11 km and 15.59 km

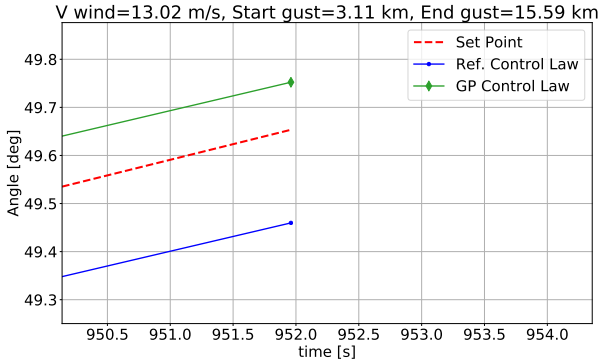


Fig. 10. Detail of the last part of the θ profile for a wind gust of 13.02 m/s applied between 3.11 km and 15.59 km

The results depicted from Fig. 7 to Fig. 10 show how the presence of a wind gust with a speed of 13.02 m/s acting between 3.11 km and 15.59 km produces a disturbance for the system which is not able to fulfill the mission (dotted line). While, with a control law evaluated through GP the system can successfully fulfill its mission (diamond marked line). Hence, also for this scenario, the GP control law proved to be necessary for the successful completion of the mission.

C. 3rd Scenario - Unknown density model

In this last scenario, the lack of knowledge in the environmental model at design time has been considered. The optimal trajectory was obtained with the density model in Eq. 2, while during the "real mission" the U.S. Standard Atmosphere Model 1962 (USSA1962) Model is used. The effect of using a different density model during the flight is shown in Fig. 12 and it is clear how the final altitude will be different than the reference without a variation of the control law.

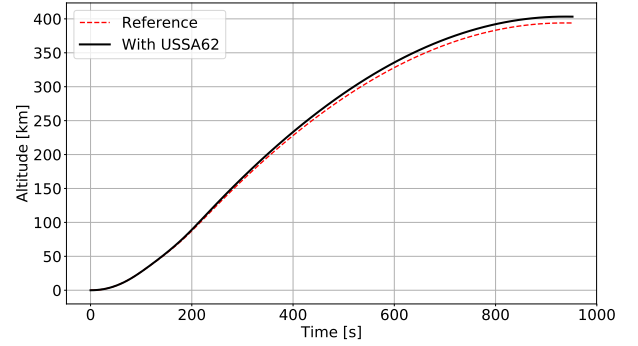


Fig. 11. Altitude profile using the two different density models. The red dashed line is the altitude profile obtained using the simplified density model while the black continuous line is the one obtained using the USSA1962 model

Figs. 12 and 13 show a comparison of the 2 different density models, where the latter shows the comparison on a semi-logarithmic scale in order to enhance the differences after 20 km. Both figures show the altitude up to 50 km because for greater altitudes, the density value is so small that is not interesting anymore. The approach chosen to define a new density model during the flight consists of performing an interpolation with a Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) as follows:

$$\rho_{new} = f(t, \rho_{old})$$

with

$$\rho_{old} = \begin{cases} \rho_{USSA1962}, & 0 \leq t \leq t_{eval} \\ \rho_{Simplified}, & t_{eval} < t \leq t_{end} \end{cases}$$

Where $\rho_{Simplified}$ is the density model in Eq. 2 and t_{eval} is the time at which the GP evaluation starts.

The idea behind this approach is to create a new density model online using the density data "measured" during the flight ($\rho_{USSA1962}$) and the data predicted by the density model known at design time ($\rho_{Simplified}$).

Figures from 14 to 17 depict the altitude and θ profile of the whole trajectory and detail of the last part of it. For the considered scenario, the density model was evaluated 3 times and each different model is depicted in Fig. 18 and 19. The evaluation of a new model was performed every time the altitude tracking error became greater than 100 m. About the aforementioned figures:

- Red dashed line: reference trajectory, obtained with the simplified density model. *Simplified model* in Fig. 18 and 19.
- Black continuous line: trajectory obtained with the USSA1962 density model and reference open loop controller. *USSA1962 model* in Fig. 18 and 19.
- Dark blue line: trajectory performed with the USSA1962 density model and reference open loop controller. It represents the first part of the flight until a variation in the tracking errors, greater than a certain threshold, is detected.
- Light blue line with diamond markers: trajectory performed using density model 0. *Model 0* in Fig. 18 and 19.
- Orange line with cross markers: trajectory performed using density model 1. *Model 1* in Fig. 18 and 19.
- Green line with dot markers: trajectory performed using density model 2. *Model 2* in Fig. 18 and 19.

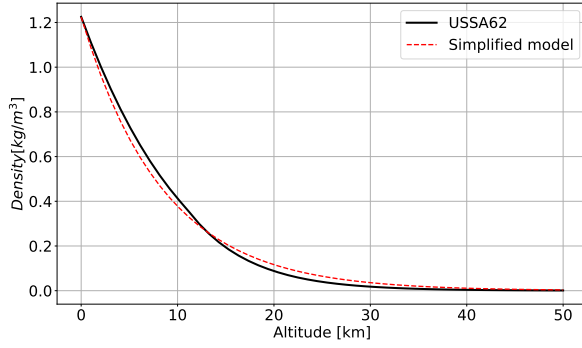


Fig. 12. Comparison between the density models up to 50 km

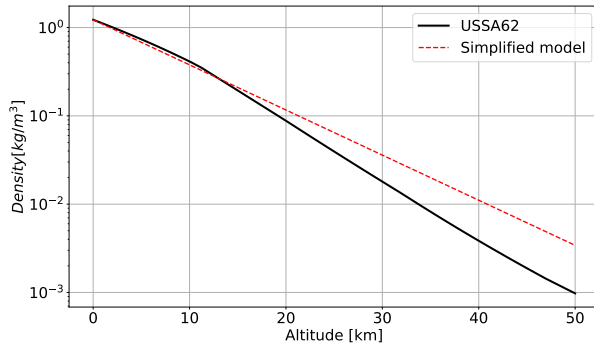


Fig. 13. Comparison between the density models up to 50 km on a semi-logarithmic scale

D. Statistics

Due to its stochastic nature, GP does not always produce satisfactory results. Hence, to prove the effectiveness of the proposed approach, some statistics are presented in Tab. II.

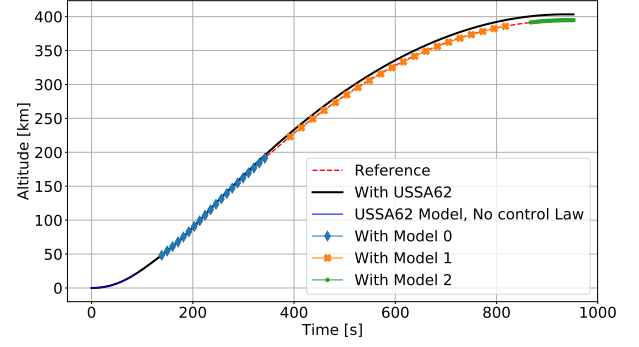


Fig. 14. Altitude profile in the 3rd scenario. The different markers on the plot represents the trajectory performed with different density models

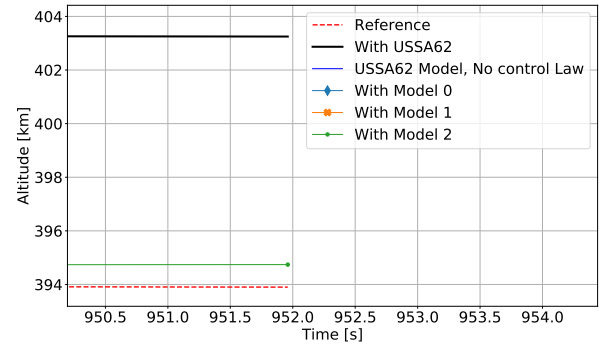


Fig. 15. Detail of the last part of the altitude profile in the 3rd scenario

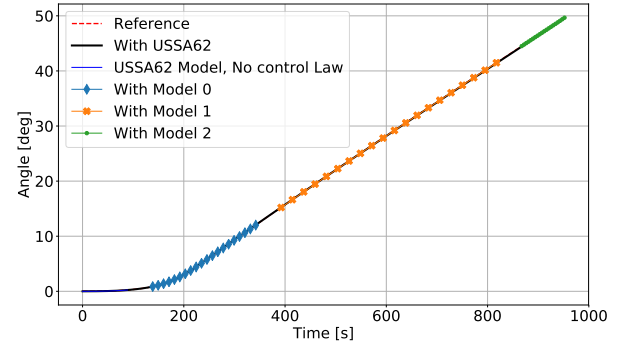


Fig. 16. θ profile in the 3rd scenario. The different markers on the plot represents the trajectory performed with different density models

Here the 3 scenarios previously discussed are presented with and without learning. More on the learning approach in the next subsection. In all 3 scenarios the mission was simulated 1000 times, with different random values of C_d , the change time, the wind speed and the altitude range for the first 2 cases. About the last case, multiple GP evaluations can be performed during the same mission. From these statistics it is clear how the GP evaluation is not always performed within the fixed

TABLE II
STATISTICS OF GP EVALUATIONS PERFORMED ON THE 3 SCENARIOS PRESENTED

	Cd Variation		Wind Gust		Density Model	
	No Learning	Learning	No Learning	Learning	No Learning	Learning
GP Evaluations	1000	1000	1000	1000	3153	3108
Fixed time interval	40 s	40 s	100 s	100 s	40 s	40 s
Min evaluation time	4.76 s	2.73 s	6.46 s	8.34 s	3.22 s	1.45 s
Max evaluation time	04h58m20s	01h18m09s	01h36m53s	26m16s	54m49s	01h01m19s
Median evaluation time	14.7 s	5.81 s	88.76 s	22.09 s	38.35 s	28.49 s
Time Constraint Success Rate	61.90%	83.10%	52.60%	90.80%	50.77%	54.89%
Range Constraint Success Rate	71.70%	73.50%	99.80%	100%	79.80%	79.60%

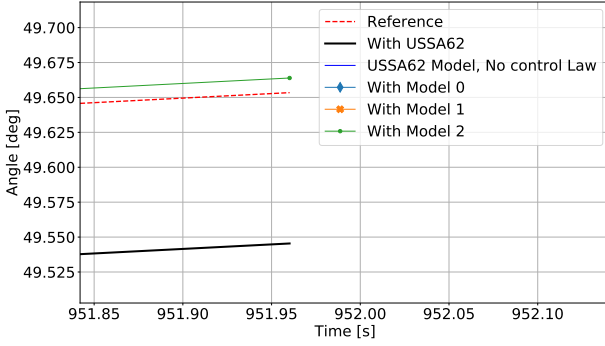


Fig. 17. Detail of the last part of θ profile in the 3rd scenario

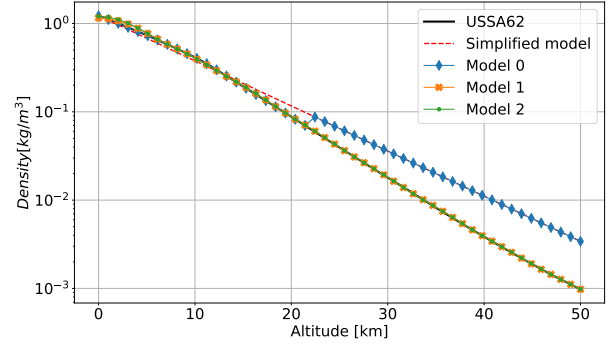


Fig. 19. Comparison between the different density models used on a semi logarithmic scale

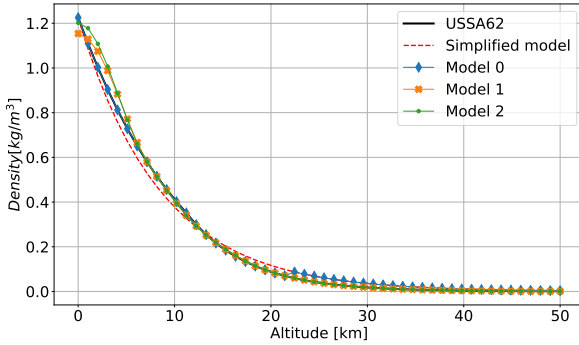


Fig. 18. Comparison between the different density models used

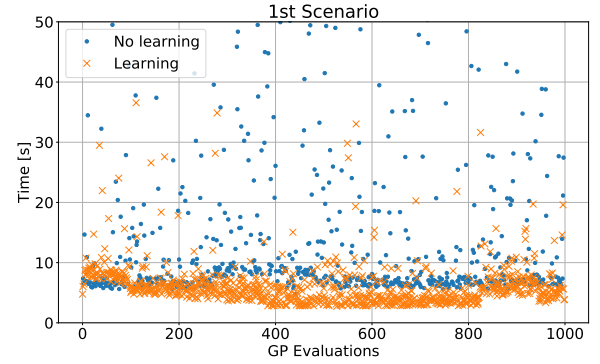


Fig. 20. Detail of time distribution for 1st scenario between 0 and 50 seconds. For this simulation the median evaluation time was of 14.7 s without learning and 5.8 s with learning

time interval, as shown by the Time Constraint Success Rate, and this is due to the high computational cost of GP.

In the last row, the Range Constraint Success Rate quantifies how many times the GP evaluations produced a satisfying control law, enabling the system to get, in its final position, within 1% range from the reference value for both altitude and θ . According to these data, the proposed approach produces good results in term of precision over 70% of the times, and it has a precision of 100% for the 2nd scenario, which might be the one closer to most real applications.

E. Learning approach

In this work, learning approach means that after one mission simulation few of the best individuals found during the current GP evolution were kept and passed to the next mission simulation. These individuals were then added to the initial population created randomly at the beginning of each mission. The maximum size of this pool of well performing individuals was of 300 individuals, while the maximum population size was set at 500 individuals. This was done in order to always guarantee that a certain portion of the population was

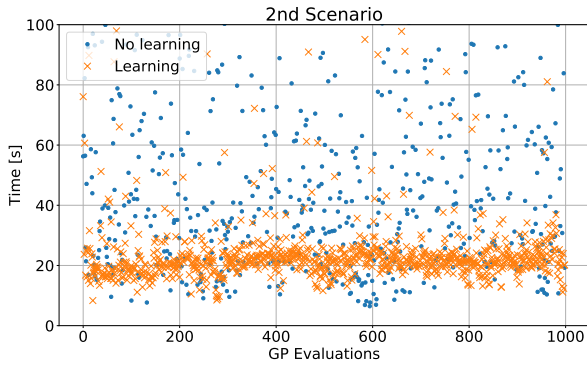


Fig. 21. Detail of time distribution for 2nd scenario between 0 and 100 seconds. For this simulation the median evaluation time was of 88.76 s without learning and 22.09 s with learning

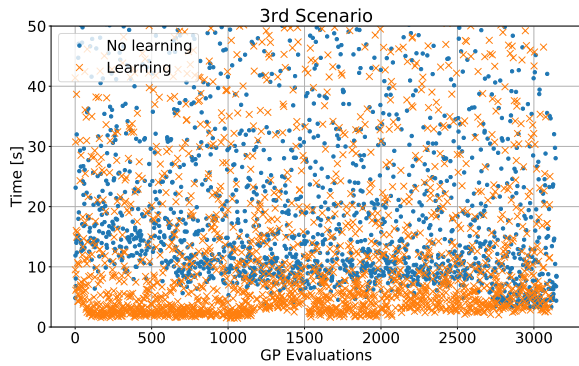


Fig. 22. Detail of time distribution for 3rd scenario between 0 and 50 seconds. For this simulation the median evaluation time was of 38.35 s without and 28.49 s with learning

generated randomly to avoid overfitting. As can be seen from Tab. II, in all 3 scenarios the learning approach increased the performances, both from the range and time constraints point of views. To better appreciate this improvements, the plots of the evaluation time against the number of evaluations are depicted from Fig. 20 to 22. These plots show the evaluation time from 0 to 50 seconds for the 1st and 3rd scenarios and from 0 to 100 s for the 2nd scenario in order to focus more on the time region that contains the median evaluation time. From these plots it is clear how the median evaluation time decreases when using the learning approach and the precise data are reported in Tab. II.

VI. CONCLUSIONS

A initial approach to Intelligent Control using Genetic Programming on 3 different failure scenarios representative of real situations has been presented. GP was used to autonomously design *online* the control law for a modified version of the Goddard Rocket problem, starting from an optimal trajectory of reference and without having any other information about the controller topology. The results obtained for the different test cases show the big potentialities of the approach and

justify a deeper research in the topic, mainly to investigate the possibilities of better trade-offs between flexibility and versatility vs computational time. The issues related to the computational costs will be mitigated by the advancements in hardware technologies. However, a better trade-off can be achieved through improvements of the GP algorithms, and hybridisation of the approach. In terms of GP, many techniques can be found in the literature that could have better performances than the standard version, like Single Node Genetic Programming or Multi Gene Genetic Programming. Some of the most promising will be considered for future tests. Another approach that can be used to improve the efficiency of the method is through hybridisation of the GP with other Artificial Intelligence (AI) techniques in order to avoid the re-evaluation of the whole control law every time using GP and just optimise it when necessary. This could be done for example by coupling GP with an Artificial Neural Network (ANN).

REFERENCES

- [1] J. Koza, M. Keane, J. Yu, F. Bennett III, and W. Mydlowec, "Automatic Creation of Human-Competitive Programs and Controllers by Means of Genetic Programming," *Genetic Programming and Evolvable Machines*, vol. 1, no. 1/2, pp. 121–164, 2000.
- [2] B. Xu and Z. K. Shi, "An overview on flight dynamics and control approaches for hypersonic vehicles," *Science China Information Sciences*, vol. 58, no. 7, pp. 1–19, 2015.
- [3] Y. chun Xie, H. Huang, Y. Hu, and G. qi Zhang, "Applications of advanced control methods in spacecrafts: progress, challenges, and future prospects," *Frontiers of Information Technology and Electronic Engineering*, vol. 17, no. 9, pp. 841–861, 2016.
- [4] K. Łapa, K. Cpałka, and A. Przybył, "Genetic programming algorithm for designing of control systems," *Information Technology and Control*, vol. 47, no. 4, pp. 668–683, 2018.
- [5] A. Bourmistrova and S. Khantsis, "Genetic Programming in Application to Flight Control System Design Optimisation," *New Achievements in Evolutionary Computation*, 2010.
- [6] R. A. Maher and M. J. Mohamed, "An Enhanced Genetic Programming Algorithm for Optimal Controller Design," *Intelligent Control and Automation*, vol. 4, pp. 94–101, 2013.
- [7] J. R. Koza, "Genetic programming as a means for programming computers by natural selection," *Statistics and Computing*, vol. 4, no. 2, pp. 87–112, 1994. [Online]. Available: <https://link.springer.com/content/pdf/10.1007%2F00175355.pdf>
- [8] C. A. Coello Coello, "Theoretical and Numerical Constraints-Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art," *Computer Methods in Applied Mechanics and Engineering* 191, pp. 1245–1287, 2002.
- [9] M. Crepinsek, S. H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM Computing Surveys*, vol. 45, no. 3, pp. 1–33, 2013.
- [10] D. Sudholt, "The Benefits of Population Diversity in Evolutionary Algorithms: A Survey of Rigorous Runtime Analyses," pp. 1–36, 2018. [Online]. Available: <http://arxiv.org/abs/1801.10087>
- [11] G. N. Saridis, "Toward the Realization of Intelligent Controls," *Proceedings of the IEEE*, vol. 67, no. 8, pp. 1115–1133, 1979.
- [12] P. J. Antsaklis, "Defining Intelligent Control. Report to the Task Force on Intelligent Control," *IEEE Control Systems Society*, pp. 1–31, 1993.
- [13] C. Wilson, F. Marchetti, M. Di Carlo, A. Riccardi, and E. Minisci, "Intelligent Control: A Taxonomy," in *2019 8th International Conference on Systems and Control (ICSC'2019)*, 2019.
- [14] R. H. Goddard, "A Method of Reaching Extreme Altitudes," *Smithsonian Miscellaneous Collections*, vol. 71, no. 2, 1919.
- [15] "Opengoddard - trajectory optimization for python," <https://github.com/istellartech/OpenGoddard>, 2017.
- [16] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, "DEAP: Evolutionary Algorithms Made Easy. Journal of Machine Learning Research," 2012.